The Use of Rippling to Automate Event-B Invariant Preservation Proofs

Yuhui Lin, Alan Bundy& Gudmund Grov

School of Informatics
University of Edinburgh

NFM 2012

Outline

In this talk we

- show that a proof technique called *rippling* is applicable to Event-B invariant proofs

- outline a novel approach combining rippling with theory formation to automate lemma discovery

Event-B invariant proofs

- Event-B is a 1st order formal modeling language
- It generates poof obligations to ensure the correctness
- Proof obligations which requires human interaction can be thousands in a industrial project

- We have observed that
  - the majority of proofs requiring automation are invariant proofs (e.g. 59% in one case study)
  - Invariant proofs typically follow a pattern where one of the assumptions is embedded in the goal, i.e. $f(x) \vdash f(\boxed{g(\boxed{x})})$

Rippling

- Rippling guides the search by writing the goal until the assumption appears as a sub-formulae, e.g. (the non-shaded part is the embedding, while the shaded part is called wave-front)

$$f(x) \vdash f(\boxed{g(\boxed{x})}) \xrightarrow{f(g(x)) = h(f(x))} f(x) \vdash \boxed{h(\boxed{f(x)})}$$

- Some Advantages:
  - allows rewrite rules in both directions without loss of termination (e.g. some distribution rules)
  - may automate many interactive proofs
- Achieved by ensure
  - the embedding is intact
  - some measure is reduced

Lemma discovery in rippling

- The key feature of rippling is the ability to automatically patch failed proofs via *critics* when required lemmas are not present
  - due to the strong expectation on the proof
- Suppose our proof is blocked at:

$$T = \mathrm{dom}(\boxed{\boxed{R} \cup S} \; ; f)$$

- We can then follow a 4 step process which discovers the missing lemma

Lemma discovery steps

1. Generate the left hand side: pick terms of blocked goals which are expected to change in the next rewriting step, e.g.

$$R \cup S \quad ; f$$

2. Conjecture right hand side scheme: we know that the right hand side must have the shape

$$F_1(R\,;\,f)(F_2 S f)$$

Where *Fn* is a 2nd order placeholders

- since the embedding must be preserved
- measure decreases

Lemma discovery steps

3. Instantiate scheme: then feed the scheme $\boxed{R \cup S} \; ; f = \boxed{F_1 (R \; ; f) (F_2 S f)}$ to IsaScheme,
   - which is a tool which discovers conjectures based on a given scheme and set of terms
   - with counter-examples checks
   - with proof attempts

4. Post filter & prove: one of the “sensible” instantiations is $(R \cup S) \; ; f = (R \; ; f) \cup (S \; ; f)$. This can be proven automatically (by Isabelle in IsaScheme), but in more complex cases the process recurses or the user must provide a proof

AI4FM

Conclusion and further work

- We have shown
  - that rippling is applicable to Event-B invariant POs
  - a new technique to help discover missing lemmas

- We are currently implementing this process in Isaplanenr.

An invariant proof using rippling

$(x \mapsto y)$	(x, y)
dom	projection
;	forward composition

$$x \in T$$
$$T = dom(R; f)$$
$$\vdash$$
$$T = dom(\boxed{(\boxed{R} \cup \{(x \mapsto y)\})}\ ; f)$$
$$T = dom(\boxed{(\boxed{R; f} \cup \{(x \mapsto y)\}; f}\ )$$
$$T = \boxed{\boxed{dom(R; f)} \cup dom(\{(x \mapsto y)\}; f)}$$
$$y \in dom(f) \vdash T = T \cup \{x\}$$
$$y \notin dom(f) \vdash T = T \cup \{\}$$